**Demo 2: Customizing the Indexing Policy for a Collection**

**Important**: Prior to getting started with this hands-on exercises, you should have already completed the **Demo1: Creating a Azure Cosmos DB Instance** hands-on exercise.

1. In a new browser window, sign in to the **Azure Portal** ([https://portal.azure.com](https://portal.azure.com/)).
2. In the *Jumpbar*, click **More Services**, locate the **Databases** section, and then click **Azure Cosmos DB**.
3. In the *Azure Cosmos DB* blade that opens, locate and click the Azure Cosmos DB account instance you created earlier in this module.
4. In the menu on the left-side of the Azure Cosmos DB account blade, locate the **Collections** section and then click the **Query Explorer** option in the menu.
5. In the *Query Explorer* blade, select the **ecommerce** database and the **customers** collection.
6. Locate the section in the current blade where you can edit the query text **New SQL Query.**

For the remainder of this hands-on exercise, we will refer to this section as the **query editor**.

1. In the query editor, replace the current query with the following query:

SELECT \*

FROM customers

ORDER BY customers.address

1. Click the **Execute Query** button.
2. In the *Results* blade, observe the results of your query.

For this hands-on exercise, we have determined that indexing the address property is inefficient and we would like to remove it from our index. The next steps in this exercise will walk us through the process of excluding a specific JSON path from the index.

1. Close the *Results* blade.
2. Locate the menu on the left-side of the **Query #** blade. In the menu, locate **Scale & Settings**
3. In the *Scale & Settings* blade, scroll down to **Settings**.
4. In the *Indexing Policy* Within the JSON editor, locate the **excludedPaths** property that has its value set to an empty array.

{

"indexingMode": "consistent",

"automatic": true,

"includedPaths": [

{

"path": "/\*",

"indexes": [

{

"kind": "Range",

"dataType": "Number",

"precision": -1

},

{

"kind": "Range",

"dataType": "String",

"precision": -1

},

{

"kind": "Spatial",

"dataType": "Point"

}

]

}

],

"excludedPaths": []

}

1. Replace the value of the **excludedPaths** property with the following array:

[

{

"path": "/address/\*"

}

]

1. Your index should now look like this:

{

"indexingMode": "consistent",

"automatic": true,

"includedPaths": [

{

"path": "/\*",

"indexes": [

{

"kind": "Range",

"dataType": "Number",

"precision": -1

},

{

"kind": "Range",

"dataType": "String",

"precision": -1

},

{

"kind": "Spatial",

"dataType": "Point"

}

]

}

],

"excludedPaths": [

{

"path": "/address/\*"

}

]

}

1. Click the **OK** button at the bottom of the blade.
2. In the *Settings* blade, click the **Save** button at the top of the blade. Wait for the collection “**Update**” operation to complete.
3. In the SQL API, locate and click the **customers** section and then click the **New SQL Query** option in the menu.
4. In the *query editor*, replace the current query with the following query:

SELECT \*

FROM customers

ORDER BY customers.address

1. Click the **Run Query** button.
2. The query should fail immediately. Scroll down to the bottom of the *Query Explorer* blade to view the error message:

*"Order-by item requires a range index to be defined on the corresponding index path."*

Because the **address** property is no longer indexed, you are unable to order the document using this property.

Source: edx.org

Microsoft: DAT221xIntroduction to NoSQL Data Solutions